

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A method comprising:
providing a system including an interface and multiple clients, wherein every client comprises code that, in being compiled, depends on information included in the interface; ~~units of compiled code, the interface including global components and each unit depending on at least one of the global components included in the interface;~~
~~dividing the interface into levels, each level including one or more of the global components, each global component being included in no more than one of the levels;~~
defining two or more levels for the interface, the two or more interface levels being defined at design time, each level corresponding to a subset of the information included in the interface;
~~generating multiple~~ two or more dependency lists for the interface, wherein each dependency list corresponds to a distinct one of the interface levels;
associating each client with a particular dependency list based on the information in the interface that the client depends on;
changing the interface and associating the change with a particular interface level and a corresponding dependency list; and
~~associating a unique one of the multiple dependency lists with each of the levels;~~
~~associating a unit with a dependency list based on the global components on which the unit depends; and~~
marking using the corresponding dependency list to mark clients for recompilation, wherein only those ~~units~~ clients associated with [[a]] the particular interface level by the corresponding dependency list are marked for recompilation based on a change to a particular global component affecting those dependency lists with relationships to a level that includes the changed global component.

2. (Cancelled)

3. (Currently Amended) The method of claim 2 further comprising recompiling the ~~unit~~ client automatically based on the marking.

4. (Currently Amended) The method of claim 3 wherein recompiling the ~~unit~~ client occurs at a subsequent usage.

5. (Original) The method of claim 4 wherein the subsequent usage is a next usage.

6. (Currently Amended) The method of claim 1 wherein marking only those client ~~units~~ associated with ~~[[a]] the particular interface level by the corresponding dependency list for recompilation based on a change to a particular global component affecting those dependency lists with relationships to a level that includes the changed global component~~ further comprises:
determining if a particular property associated with the level has changed; and
marking the ~~unit~~ client for recompilation only if a particular property has changed.

7. (Original) The method of claim 1 wherein dividing the interface into levels further comprises assigning an arbitrary number of levels to the interface.

8. (Original) The method of claim 1 wherein dividing the interface into levels includes assigning a level based on a dependency on all levels of the interface.

9. (Original) The method of claim 8 further comprising recompiling a client assigned to the level based on a strong dependency on the whole interface after each change to the interface.

10. (Original) The method of claim 1 wherein dividing the interface into levels further comprises assigning a level based on a dependency on an interface component.

11. (Currently Amended) The method of claim 10 further comprising, recompiling a ~~unit~~ client assigned to the level based on a dependency on an interface component after each change to the component.

12. (Original) The method of claim 11 wherein the change to the component includes a name change.

13. (Original) The method of claim 11 wherein the change to the component includes a deletion of a component.

14. (Original) The method of claim 11 wherein the change to the component includes a layout change.

15. (Original) The method of claim 1 wherein dividing the interface into levels includes assigning a level based on a reference to the interface.

16. (Original) The method of claim 15 wherein the client depends on the existence of the interface.

17. (Original) The method of claim 1 further comprising associating indirect clients with a level.

18. (Currently Amended) The method of claim 17 wherein the indirect clients are associated with a lower level than the ~~units~~ clients.

19. (Original) The method of claim 1 wherein the dependency list is automatically managed by the system.

20. (Currently Amended) A computer program product, tangibly embodied in a machine-readable storage device, ~~for executing instructions on a processor,~~ the computer program product being operable to cause a machine to:

provide a system including an interface and multiple clients, wherein every client comprises code that, in being completed, depends on information included in the interface;~~units of compiled code, the interface including global components and each unit depending on at least one of the global components included in the interface;~~

~~divide the interface into levels, each level including one or more of the global components, each global component being included in no more than one of the levels;~~

define two or more levels for the interface, the two or more interface levels being defined at design time, each level corresponding to a subset of the information included in the interface;

generate multiple two or more dependency lists for the interface, wherein each dependency list corresponds to a distinct one of the interface levels;

associate each client with a particular dependency list based on the information in the interface that the client depends on;

change the interface and associate the change with a particular interface level and a corresponding dependency list; and

~~associate a unique one of the multiple dependency lists with each of the levels;~~

~~associate a unit with a dependency list based on the global components on which the unit depends; and~~

mark use the corresponding dependency list to mark clients for recompilation, wherein only those ~~units~~ clients associated with [[a]] the particular interface level by the corresponding dependency list are marked for recompilation based on a change to a particular global component affecting those dependency lists with relationships to a level that includes the changed global component.

21. (Original) The computer program product of claim 20 further comprising, instructions to cause a machine to recompile the client automatically based on the marking.

22-26. (Canceled)

27. (Currently Amended) A system comprising:

a primary system including an interface and multiple clients, wherein every client comprises code that, in being completed, depends on information included in the interface ~~and multiple units of compiled code, the interface including global components, and each unit depending on at least one of the global components in the interface;~~

a recompilation system including a processor and a memory storing a computer program product that includes instructions operable to cause the processor to:

define two or more levels for the interface, the two or more interface levels being defined at design time, each level corresponding to a subset of the information included in the interface;

~~divide the interface into levels, each level including one or more of the global components, each global component being included in no more than one of the levels;~~

generate ~~multiple~~ two or more dependency lists for the interface, wherein each dependency list corresponds to a distinct one of the interface levels;

associate each client with a particular dependency list based on the information in the interface that the client depends on;

change the interface and associate the change with a particular interface level and a corresponding dependency list; and

~~associate a unique one of the multiple dependency lists with each of the levels;~~

~~associate a unit with a dependency list based on the global components on which the unit depends; and~~

~~mark~~ use the corresponding dependency list to mark clients for recompilation, wherein only those ~~units~~ clients associated with [[a]] the particular interface level by the corresponding dependency list are marked for recompilation ~~based on a change to a particular global component affecting those dependency lists with relationships to a level that includes the changed global component.~~

28. (Currently Amended) The system of claim 27 in which the computer program product stored in the memory of the recompilation system further includes instructions operable

to cause the processor of the recompilation system to automatically recompile only those ~~units~~
clients that are marked.

29. (Cancelled)

30. (Currently Amended) The system of claim 27 in which the computer program product stored in the memory of the recompilation system further includes instructions operable to cause the processor of the recompilation system to:

determine if a property associated with the level has changed, and
mark the ~~unit~~client for recompilation only if a property has changed.

31-33. (Canceled)